PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

THE ACM DIGITAL LIBRARY

Feedback

combine null check with barrier
Terms used: combine null check with barrier

Found

Sort results by: relevance

Display results: expanded form

Save results to a Binder

□ Open results in a new window

Refine these results with Adv

Try this search in The ACM G

Results 1 - 20 of 192   Result page: 1 2 3 4 5 6 7 8 9 10 next >>

1 Speculative optimization using hardware-monitored guarded regions for java virtual machines
Lixin Su, Mikko H. Lipasti
June 2007 VEE '07: Proceedings of the 3rd international conference on Virtual execution environments
Publisher: ACM
Full text available: pdf(357.43 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 4, Downloads (12 Months): 124, Citation Count: 0

Aggressive dynamic optimization in high-performance Java Virtual Machines can be hampered by language features like Java's exception model, which requires precise detection and handling of program-generated exceptions. Furthermore, the compile-time overhead ...

Keywords: java, precise exceptions, speculative processors, transactional memory, virtual machines

2 Flow-insensitive type qualifiers
Jeffrey S. Foster, Robert Johnson, John Kodumal, Alex Aiken
November 2006 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 28 Issue 6
Publisher: ACM
Full text available: pdf(910.93 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 7, Downloads (12 Months): 79, Citation Count: 4

We describe flow-insensitive type qualifiers, a lightweight, practical mechanism for specifying and checking properties not captured by traditional type systems. We present a framework for adding new, user-specified type qualifiers to programming languages ...

Keywords: Type qualifiers, const, constraints, security, static analysis, taint, types

3 Write barrier elision for concurrent garbage collectors
Martin T. Vechev, David F. Bacon
October 2004 ISMM '04: Proceedings of the 4th international symposium on Memory management

Ads b
Goo
Sear
diffe
& mc
types
googl
Wall
Jour
Wall
Free
Wee
www.
Jour
Find
Rese
Sour
For F
www.
Jour
2016
The
peop
Beca
Inc
www.j

Publisher: ACM

Full text available: pdf(490.73 KB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 7, Downloads (12 Months): 46, Citation Count: 2

Concurrent garbage collectors require write barriers to preserve consistency, but these barriers impose significant direct and indirect costs. While there has been a lot of work on optimizing write barriers, we present the first study of their elision ...

Keywords: concurrent garbage collection, write barrier

4 Design and evaluation of dynamic optimizations for a Java just-in-time compiler

Toshio Suganuma, Toshiaki Yasue, Motohiro Kawahito, Hideaki Komatsu, Toshio Nakatani

July 2005 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 27 Issue 4

Publisher: ACM

Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 24, Downloads (12 Months): 201, Citation Count: 3

The high performance implementation of Java Virtual Machines (JVM) and Just-In-Time (JIT) compilers is directed toward employing a dynamic compilation system on the basis of online runtime profile information. The trade-off between the compilation overhead ...

Keywords: JIT compiler, Recompilation, adaptive optimization, code specialization, dynamic compilation, profile-directed method inlining

5 Enforcing isolation and ordering in STM

Tatiana Shpeisman, Vijay Menon, Ali-Reza Adl-Tabatabai, Steven Balensiefer, Dan Grossman, Richard L. Hudson, Katherine F. Moore, Bratin Saha

June 2007 PLDI '07: Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation

Publisher: ACM

Full text available: pdf(257.39 KB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 23, Downloads (12 Months): 327, Citation Count: 1

Transactional memory provides a new concurrency control mechanism that avoids many of the pitfalls of lock-based synchronization. High-performance software transactional memory (STM) implementations thus far provide *weak atomicity*: Accessing shared ...

Keywords: code generation, compiler optimizations, escape analysis, isolation, ordering, strong atomicity, transactional memory, virtual machines, weak atomicity

6 Enforcing isolation and ordering in STM

Tatiana Shpeisman, Vijay Menon, Ali-Reza Adl-Tabatabai, Steven Balensiefer, Dan Grossman, Richard L. Hudson, Katherine F. Moore, Bratin Saha

June 2007 ACM SIGPLAN Notices, Volume 42 Issue 6

Publisher: ACM

Full text available: pdf(257.39 KB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 23, Downloads (12 Months): 327, Citation Count: 1

Transactional memory provides a new concurrency control mechanism that avoids many of the pitfalls of lock-based synchronization. High-performance software transactional memory (STM) implementations thus far provide *weak atomicity*: Accessing shared ...

Keywords: code generation, compiler optimizations, escape analysis, isolation, ordering, strong atomicity, transactional memory, virtual machines, weak atomicity

7 Overlooking roots: a framework for making nondeferred reference-counting garbage collection fast

Pramod G. Joisha

October 2007 ISMM '07: Proceedings of the 6th international symposium on Memory management

Publisher: ACM

Full text available: pdf(322.01 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 2, Downloads (12 Months): 55, Citation Count: 0

Numerous optimizations exist for improving the performance of nondeferred reference-counting (RC) garbage collection. Their designs are ad hoc, intended to exploit different count removal opportunities. This paper shows that many of these optimizations ...

Keywords: reference counting, static analysis

8 Fault-safe code motion for type-safe languages

Brian R. Murphy, Vijay Menon, Florian T. Schneider, Tatiana Shpeisman, Ali-Reza Adl-Tabatabai

April 2008 CGO '08: Proceedings of the sixth annual IEEE/ACM international symposium on Code generation and optimization

Publisher: ACM

Full text available: pdf(421.78 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 27, Downloads (12 Months): 27, Citation Count: 0

Compilers for Java and other type-safe languages have historically worked to overcome overheads and constraints imposed by runtime safety checks and precise exception semantics. We instead exploit these safety properties to perform code motion optimizations ...

Keywords: code motion, intermediate representations, partial redundancy elimination, safe code motion, safety dependences, scheduling, speculative code motion

9 GNAT: on the road to Ada 2005

Javier Miranda, Edmond Schonberg

December 2004 ACM SIGAda Ada Letters, Volume XXIV Issue 4

Publisher: ACM

Full text available: pdf(179.24 KB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 0, Downloads (12 Months): 29, Citation Count: 1

The GNAT Development Team is directly involved with the Ada 2005 effort, both participating in the Ada Rapporteur Group (ARG), and implementing and testing the new features proposed for the language revision. In this paper we summarize the Ada 2005 issues ...

Keywords: Ada 2005, GNAT, compiler, front-end

10 GNAT: on the road to Ada 2005
Javier Miranda, Edmond Schonberg
November 2004 SIGAda '04: Proceedings of the 2004 annual ACM SIGAda international conference on Ada: The engineering of correct and reliable software for real-time & distributed systems using Ada and related technologies
Publisher: ACM
Full text available: pdf(179.24 KB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 0, Downloads (12 Months): 29, Citation Count: 1

The GNAT Development Team is directly involved with the Ada 2005 effort, both participating in the Ada Rapporteur Group (ARG), and implementing and testing the new features proposed for the language revision. In this paper we summarize the Ada 2005 issues ...

Keywords: Ada 2005, GNAT, compiler, front-end

11 Removing redundancy via exception check motion
Vijay Sundaresan, Mark Stoodley, Pramod Ramarao
April 2008 CGO '08: Proceedings of the sixth annual IEEE/ACM international symposium on Code generation and optimization
Publisher: ACM
Full text available: pdf(252.57 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 25, Downloads (12 Months): 25, Citation Count: 0

Partial redundancy elimination aims to reduce the number of times an expression is computed more than once. The traditional Lazy Code Motion (LCM) algorithm formulated by Knoop, Ruthing and Steffen, through its reliance on unordered bit vectors, is severely ...

Keywords: java, just-in-time compilation, partial redundancy elimination.

12 PolyD: a flexible dispatching framework
Antonio Cunei, Jan Vitek
October 2005 ACM SIGPLAN Notices, Volume 40 Issue 10
Publisher: ACM
Full text available: pdf(1.70 MB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 6, Downloads (12 Months): 34, Citation Count: 0

The standard dispatching mechanisms built into programming languages are sometimes inadequate to the needs of the programmer. In the case of Java, the need for more flexibility has led to the development of a number of tools, including visitors and multi-method ...

Keywords: Java, dispatching, multimethods, visitor pattern

13 PolyD: a flexible dispatching framework
Antonio Cunei, Jan Vitek
October 2005 OOPSLA '05: Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications
Publisher: ACM

Full text available: pdf(1.70 MB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 6, Downloads (12 Months): 34, Citation Count: 0

The standard dispatching mechanisms built into programming languages are sometimes inadequate to the needs of the programmer. In the case of Java, the need for more flexibility has led to the development of a number of tools, including visitors and multi-method ...

Keywords: Java, dispatching, multimethods, visitor pattern

14 Secure virtual architecture: a safe execution environment for commodity operating systems

John Criswell, Andrew Lenharth, Dinakar Dhurjati, Vikram Adve

October 2007 SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles

Publisher: ACM

Full text available: pdf(383.30 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 33, Downloads (12 Months): 299, Citation Count: 0

This paper describes an efficient and robust approach to provide a *safe execution environment* for an entire operating system, such as Linux, and all its applications. The approach, which we call *Secure Virtual Architecture* (SVA), defines ...

Keywords: compiler, memory safety, operating systems, security, type safety, typed assembly language, virtual machine

15 Secure virtual architecture: a safe execution environment for commodity operating systems

John Criswell, Andrew Lenharth, Dinakar Dhurjati, Vikram Adve

October 2007 ACM SIGOPS Operating Systems Review, Volume 41 Issue 6

Publisher: ACM

Full text available: pdf(383.30 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 33, Downloads (12 Months): 299, Citation Count: 0

This paper describes an efficient and robust approach to provide a *safe execution environment* for an entire operating system, such as Linux, and all its applications. The approach, which we call *Secure Virtual Architecture* (SVA), defines ...

Keywords: compiler, memory safety, operating systems, security, type safety, typed assembly language, virtual machine

16 Compiler optimizations for nondeferred reference: counting garbage collection

Pramod G. Joisha

June 2006 ISMM '06: Proceedings of the 5th international symposium on Memory management

Publisher: ACM

Full text available: pdf(220.00 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 1, Downloads (12 Months): 68, Citation Count: 0

Reference counting is a well-known technique for automatic memory management, offering unique advantages over other forms of garbage collection. However, on account of the high costs associated with the maintenance of up-to-date tallies of references ...

Keywords: reference counting, static analyses

17 Xoc, an extension-oriented compiler for systems programming
Russ Cox, Tom Bergan, Austin T. Clements, Frans Kaashoek, Eddie Kohler
March 2008 ASPLOS XIII: Proceedings of the 13th international conference on Architectural support for programming languages and operating systems
Publisher: ACM
Full text available: pdf(208.34 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 31, Downloads (12 Months): 89, Citation Count: 0

Today's system programmers go to great lengths to extend the languages in which they program. For instance, system-specific compilers find errors in Linux and other systems, and add support for specialized control flow to Qt and event-based programs. ...

Keywords: extension-oriented compilers

18 Design of the Java HotSpot™ client compiler for Java 6
Thomas Kotzmann, Christian Wimmer, Hanspeter Mössenböck, Thomas Rodriguez, Kenneth Russell, David Cox
May 2008 ACM Transactions on Architecture and Code Optimization (TACO), Volume 5 Issue 1
Publisher: ACM
Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 0, Downloads (12 Months): 0, Citation Count: 0

Version 6 of Sun Microsystems' Java HotSpot™ VM ships with a redesigned version of the client just-in-time compiler that includes several research results of the last years. The client compiler is at the heart of the VM configuration used by default ...

Keywords: Java, compiler, deoptimization, intermediate representation, just-in-time compilation, optimization, register allocation

19 High-level real-time programming in Java
David F. Bacon, Perry Cheng, David Grove, Michael Hind, V. T. Rajan, Eran Yahav, Matthias Hauswirth, Christoph M. Kirsch, Daniel Spoonhower, Martin T. Vechev
September 2005 EMSOFT '05: Proceedings of the 5th ACM international conference on Embedded software
Publisher: ACM
Full text available: pdf(341.79 KB) Additional Information: full citation, abstract, references, cited by, index terms

Bibliometrics: Downloads (6 Weeks): 7, Downloads (12 Months): 52, Citation Count: 2

Real-time systems have reached a level of complexity beyond the scaling capability of the low-level or restricted languages traditionally used for real-time programming.While Metronome garbage collection has made it practical to use Java to implement ...

Keywords: WCET, allocation, scheduling, tasks, visualization

20 Automatic feedback-directed object inlining in the java hotspot™ virtual machine
Christian Wimmer, Hanspeter Mössenböck
June 2007 VEE '07: Proceedings of the 3rd international conference on Virtual execution environments

Publisher: ACM

Full text available: pdf(341.49 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 5, Downloads (12 Months): 118, Citation Count: 0

Object inlining is an optimization that embeds certain referenced objects into their referencing object. It reduces the costs of field accesses by eliminating unnecessary field loads. The order of objects in the heap is changed in such a way that ...

Keywords: cache, garbage collection, java, just-in-time compilation, object colocation, object inlining, optimization, performance

Results 1 - 20 of 192 Result page: 1 2 3 4 5 6 7 8 9 10 next >>

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2008 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player